

REMARKS

Claims 1-14 are pending in this application. Claims 4, 6, 8 and 10 have been amended to overcome the claim objections and the 35 U.S.C. §112 rejection. Reconsideration of the pending claims in view of the amendments and the following remarks is respectfully requested. The various issues raised by the Office Action are addressed next in the order in which they appear in the Office Action.

Claim Objections

The Office Action, in paragraph 3, objected to claims 4-9 as being in improper form because a multiple dependent claim can not depend on another multiple dependent claim. Accordingly, original multiple dependent claims 4, 6 and 8 are amended to overcome this objection. Specifically, claim 4 is amended so that it does not depend on multiple dependent claim 3. Moreover, amended claim 6 depends on claim 4 only. Amended claim 8 depends on claim 6 only. Applicants respectfully submitted that amended claims 4-9 are now in proper form because there no longer exists any multiple dependent claim that depends on another multiple dependent claim.

112 Rejection

The Office Action, in paragraph 4, rejected claim 10 under 35 U.S.C. §112, second paragraph, as being indefinite in that it fails to point out what is included or excluded by the claim language. Accordingly, claim 10 is amended to recite “a software analysis tool substantially as described with reference to Figs. 1-18.” Applicants respectfully request that the 112 rejection be withdrawn because amended claim 10 specifically refers to Figs. 1-18.

102 Rejections

The Office Action, in paragraphs 5-6, rejected claims 1 and 10-14 under 35 U.S.C. 102 (b) as being anticipated by Brotsky (US Patent Number 5,490,246) (“Brotsky”). Applicants respectfully submit that claims 1 and 10-14 are not anticipated by Brotsky for at least the following reasons.

First, Brotsky is directed to a graphical editor where the user creates a directed graph (the ACG) in order to describe how an image is constructed out of one or more figural elements such as graphical images. See col. 5, ln. 4-20. The ACG taught by Brotsky has three types of nodes: source nodes, transfer nodes, and sink nodes. A source node represents source data, such as a graphical image segment, that needs to be edited upon. One or more source node in the direct graph may be connected to a transfer node. A transfer node represents an operation, such as superimposing one image onto another, that is performed upon the inputting source data. A transfer node may be connected to another transfer node, or to a sink node. A sink node represents the output image resulting from the operations described by the transfer nodes. Images are created by traversing the ACG from the source nodes to the transfer nodes and to the sink nodes. See col. 6, ln 37-col. 7, ln 35. Thus, Brotsky's ACG only describes the input-output relationship between image data. Brotsky does not teach, disclose, or suggest analyzing software entities, such as Java or C++ code packages. Nor does Brotsky teach, disclose, or suggest analyzing the dependency relationship, such as inclusion, association, or inheritance, between components of software or other systems. Moreover, Brotsky's ACG is a flat graph in that it does not contain any node that further contains a child ACG. Furthermore, Brotsky does not teach, disclose, or even suggest folding a group of nodes into a single node or unfolding a node in the ACG to display the child graph contained in the node.

By contrast, pending claim 1 expressly requires “a software analysis tool comprising: means for converting software entities and their relationships into a graph having a structure of nodes interconnected by edges”. Pending claim 10 recites “a software analysis tool substantially as described with reference to Figs. 1-18”. Among the figures referred to by claim 10, figure 1 schematically describes a software analysis tool that converts software code and their relationships into a graph. Pending claims 11 and 13-14 all recite “a dependency analysis system”. Moreover, pending claim 11 further requires “a connection class for instantiating connection objects in memory representing dependencies between aspects of an analyzed system”. Pending claim 13 further requires “a graph model data structure for storing dependency information derived through the abstraction layer from third-party tools”. Pending claim 14 further requires unfolding using “a user interface ... causing the rendering system to replace the displayed node with one or more child nodes in response to the user action”.

Accordingly, it is respectfully submitted that Brotsky does not anticipate pending claims 1, 10-11 and 13-14 because Brotsky does not teach, disclose, or suggest analyzing software entities or performing dependency analysis.

Pending claim 12 depends from claim 11. Therefore, it is respectfully submitted that claim 12 is not anticipated by Brotsky for at least the reason associated with claim 11.

Accordingly, it is respectfully requested that all of the Section 102 rejections be withdrawn.

103 rejections

The Office Action, in paragraph 7, rejected claims 2-3 under 35 U.S.C. 103 (a) as being unpatentable over Brotsky in view of Guy E. Blelloch, “Provably Efficient Scheduling for Languages with fine-Grained Parallelism” (“Blelloch”). It is respectfully submitted that claims 2-3 are patentable over Brotsky in view of Blelloch for at least the following reasons.

Blelloch teaches the dynamic scheduling of tasks performed by parallel algorithms during program execution. The scheduler taught by Blelloch is a “dynamic” run-time scheduler in that it allocates computing resources in a step by step fashion as the structure of the algorithm is revealed during its execution. See, e.g., Abstract on pg. 281-282. In contrast, the software analysis tool defined by claim 1 is a “static” tool in that it converts software entities and their relationships into graphs without executing those software entities. Moreover, Blelloch does not teach, disclose, or even suggest “an editor comprising means for allowing a user to edit the graph”. In fact, Blelloch would have no desire to let a user to interfere with the execution of the dynamic task scheduler. Therefore, the combined teachings of Brotsky and Blelloch do not render claims 2-3 unpatentable since claims 2-3 depend from patentable independent claim 1 and claim 1 expressly requires a software analysis tool that converts software entities to graphs, comprising an editor comprising means for allowing a user to edit the graph.

Furthermore, it is respectfully submitted that Blelloch does not teach, disclose, or even suggest bi-directionally folding and unfolding a graph between meta and child levels, as required by claims 2-3. Specifically, the disclosure at pg. 301, section 5.4.2 of Blelloch only teaches “dynamically unfolding DAGs” (Directed Acyclic Graphs)

whose nodes and edges are dynamically created as the program is executed. In contrast, the term “unfolding” in this application refers to the removal of a meta node on a graph and replacing it with its contained child graph. Therefore, “unfolding DAGs” as taught by Blelloch is entirely different from “unfolding a meta node” as taught by this application.

The Office Action, in paragraph 8, rejected claims 4-9 under 35 U.S.C. 103 (a) as being unpatentable over Brotsky in view of Blelloch, further in view of Perttunen (US Patent Number 6,359,635) (“Perttunen”). It is respectfully submitted that these claims are patentable over Brotsky, Blelloch and Perttunen for at least the following reasons.

Perttunen teaches the presentation of categorized information, such as organization charts, in the form of graphs and charts. See, e.g., col. 1, lns. 7-9. Like Brotsky and Blelloch, Perttunen does not teach, disclose, or suggest analyzing software entities or performing dependency analysis. Therefore, the combined teachings of Brotsky, Blelloch and Perttunen do not render claims 4-9 unpatentable since claims 4-9 ultimately depend from patentable independent claim 1 and claim 1 expressly requires a software analysis tool that converts software entities to graphs.

Accordingly, it is respectfully requested that all of the Section 103 rejections be withdrawn.



CONCLUSION

In light of the above, it is respectfully submitted that the present application is in condition for allowance. Favorable disposition is respectfully requested. Should the Examiner have any questions or comments concerning this submission, or any aspect of the application, the Examiner is respectfully invited to call the undersigned at the phone number listed below.

No fee other than the time extension fee is believed due at this time. Should any fees be required, please charge such fees to Pennie & Edmonds LLP Account No. 16-1150.

Respectfully submitted,

 Reg. No. 51,796

 Barry D. Rein (Reg. No. 22,411)

PENNIE & EDMONDS LLP
1155 Avenue of the Americas
New York, New York 10036
(212) 790-9090